Listing of the Claims:

This listing of claims will replace all prior versions, and listing of claims in the application:

1. (Previously Presented) In a computing environment, a macro virtual machine loop instruction for execution by a virtual machine,

wherein said macro virtual machine loop instruction represents a conventional sequence of Bytecode instructions in a programming loop that can be executed by said virtual machine, said conventional sequence of Bytecode instructions including a conventional conditional flow control Bytecode;

wherein said macro instruction is a single virtual machine instruction that can effectively replace said conventional sequence of Bytecode instructions and can be executed by said virtual machine operating in said computing environment,

wherein when said macro instruction is executed, the operations that are performed by said conventional sequence of Bytecode instructions are performed; and

wherein said macro virtual machine loop instruction is generated and loaded into said virtual machine instead of said conventional sequence of Bytecode instructions during the Bytecode verification prior to execution time.

2-3. (Cancelled)

4. (Previously Presented) A macro instruction as recited in claim 1, wherein said virtual machine internally represents instructions as a pair of streams.

5. (Previously Presented) A macro instruction as recited in claim 4,

wherein said pair of streams includes a code stream and a data stream,

wherein said code stream is suitable for containing a code portion of said macro instruction, and

wherein said data stream is suitable for containing a data portion of said macro instruction.

6. (Previously Presented) A macro instruction as recited in claim 5,

wherein said macro instruction is generated only when said virtual machine determines that said macro instruction should replace said sequence.

7. (Original) A Java macro instruction as recited in claim 6, wherein said determination is made based on a predetermined criteria.

8. (Previously Presented) A macro instruction as recited in claim 7, wherein said predetermined criteria is whether said sequence has been repeated more than a predetermined number of times.

9. (Previously Presented) A macro virtual machine loop instruction as recited in claim 1,

wherein said virtual machine loop instruction is in a reduced set of virtual machine instructions suitable for execution in a virtual machine, the reduced set of virtual machine instructions representing a full set of conventional Bytecode executable instructions that are also suitable for execution in the virtual machine,

wherein the reduced set of the virtual machine instructions consists of a number of virtual machine instructions which is less than the number of instructions in the full set of conventional Bytecode executable instructions, and

wherein every one of the instructions in the full set of conventional Bytecode executable instructions can be represented by at least one of the virtual machine instructions in the virtual machine instruction set.

10-20 (Cancelled)

21. (Previously Presented) A method for executing a programming loop by a virtual machine, said method comprising:

receiving during Bytecode verification at load time a plurality of conventional Bytcodes instructions which can be executed by said virtual machine;

determining during Bytecode verification at load time whether said plurality of conventional Bytcodes instructions includes a predetermined first sequence of Bytecode instructions that performs a programming loop, wherein said first sequence includes a conventional conditional flow control Bytecode;

generating at load time a single macro virtual machine loop instruction that can effectively replace said first sequence of Bytecode instructions when said determining determines that said plurality of conventional Bytcodes instructions includes said first sequence of Bytecode instructions that performs a programming loop;

loading at load time said single macro virtual machine loop instruction into said virtual machine instead of said first sequence of Bytecode instructions; and

executing at runtime said single macro virtual machine loop instruction by said virtual machine, thereby allowing said first sequence of Bytecode instructions to be performed by said macro virtual machine loop instruction to perform said programming loop.

22. (Previously Presented) A method as recited in claim 21, wherein said virtual machine internally represents instructions as a pair of streams.

23. (Previously Presented) A method as recited in claim 22, wherein said pair of streams includes a code stream and a data stream,

wherein said code stream is suitable for containing a code portion of said macro instruction, and

wherein said data stream is suitable for containing a data portion of said macro instruction.

24. (Previously Presented) A method as recited in claim 21, wherein said macro virtual machine loop instruction is generated only when said virtual machine determines that said first sequence has been repeated more than a predetermined number of times.

25. (Previously Presented) A virtual machine for executing instructions that include a programming loop, wherein said virtual machine is capable of:

receiving during Bytecode verification at load time a plurality of conventional Bytcodes instructions which can be executed by said virtual machine;

determining during Bytecode verification at load time whether said plurality of conventional Bytcodes instructions includes a predetermined first sequence of Bytecode instructions that performs a programming loop, wherein said first sequence includes a conventional conditional flow control Bytecode;

generating at load time a single macro virtual machine loop instruction that can effectively replace said first sequence of Bytecode instructions when said determining determines that said plurality of conventional Bytcodes instructions includes said first sequence of Bytecode instructions that performs a programming loop;

loading at load time said single macro virtual machine loop instruction into said virtual machine instead of said first sequence of Bytecode instructions; and

executing at runtime said single macro virtual machine loop instruction by said virtual machine, thereby allowing said first sequence of Bytecode instructions to be performed by said macro virtual machine loop instruction to perform said programming loop.

26. (Previously Presented) A virtual machine as recited in claim 25, wherein said virtual machine internally represents instructions as a pair of streams.

27. (Previously Presented) A virtual machine as recited in claim 26, wherein said pair of streams includes a code stream and a data stream,

wherein said code stream is suitable for containing a code portion of said macro instruction, and

wherein said data stream is suitable for containing a data portion of said macro instruction.

28. (Previously Presented) A virtual machine as recited in claim 25, wherein said macro virtual machine loop instruction is generated only when said virtual machine determines that said sequence has been repeated more than a predetermined number of times.